University of Dhaka

Department of Computer Science and Engineering

Project Report:

Fundamentals of Programming Lab(CSE-1211)

Project Name:

DX-BALL

Team Members:

1. Diponker Roy , Roll-28.
2. Tahshina Islam Munni , Roll-26.
3. Mohammad Asif , Roll-54.

Introduction:

Dx ball is a single player game.The main goal is breaking all the bricks without losing the ball. As it is a common Breakout-style game, the object is to clear the screen of all bricks in order to advance to the next level. The player controls a paddle at the bottom of the screen by using the mouse/keyboard, keeping one ball in play by bouncing it into a field of bricks to clear them. Some bricks will take multiple hits before they clear, while other bricks will appear to be unbreakable. progress by enhancing the paddle with guns . If the player misses the ball in play, a paddle will be lost. The game will end once all spare paddles have been lost, or after the selected level has been completed.The game keeps the scores of a player, if a player can score a point which is in the range of maximum scores of all time.

Objective:

Our major goal is to use the SDL (Simple DirectMedia Layer) library and C/C++ language to construct a straightforward graphical game project in the real world. However, in order to create and use it, we wanted to increase our C/C++ understanding.

Project Features:

The features of the project DX-BALL can be split into two categories. There are two types: UI (User Interface) features and game-play features. The panels, pages, and other visual components like buttons and icons that make up the user

interface (UI) features allow players to interact with the game. Game-play features, on the other hand, are those that may be used, accomplished, or imagined while playing the game. To enhance user experience and make the game more fascinating and versatile, many features have been introduced.
The attributes are:

• User Interface Features:

1. Main Menu Features:

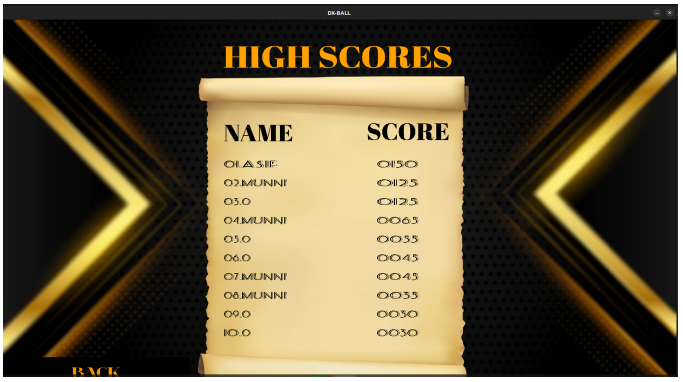
This page will be shown by the user both when they start the game and when they finish it. The player can access this page by clicking on the icons for New Game, High Score, Options, Help, and Exit. Images and names of various power-up features are displayed in the lower corner of the page. There will be music if the "Music" option is selected in the "Option" menu.Figure 1: Main Menu Page The player can select from this page to begin a new game, view the previous high scores, go to the option menu for controlling the game's noises, select the help section for instructions and control features, and select the exit menu to end the game. The following is a description of the main menu page's icons:

1.1 New Game:

This option will take a player to the level-1 of the game. All the previous game data will be lost after that, and the player has to start again. To start a new game player has to select the “New game” icon.

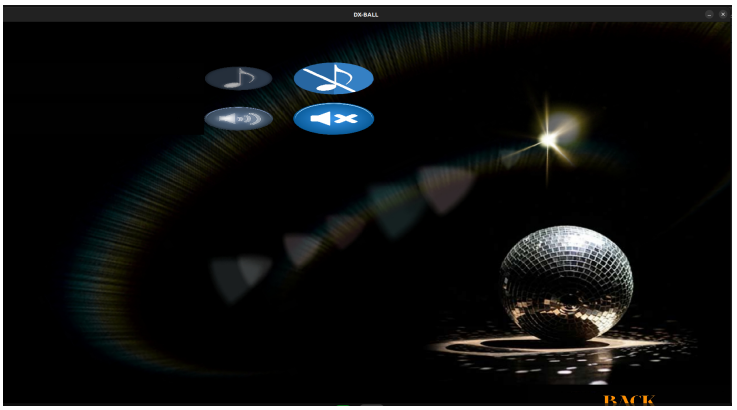
1.2 High Score:

This selection will display the top 20 rankings for prior tests games in order of difficulty. First page will display the first 10 highest rankings after selecting the "Next," score button, a second page will show up and display the other ten top scores.



1.3 Option:

This icon will take the player to the page from where the player will be able to turn ON/OFF the game music and also turn ON/OFF the sound (In game effect sounds. Such as:paddle and bar collision,ball and bricks collision).



1.4 End Game:

The player will see a confirmation notice with the choices "YES" and "NO" after selecting the option.

The player will return to the main menu screen if they select "NO" as opposed to "YES," which will end the game.

1.5 Instruction:

After choosing this option the player will see a page with instructions for the game.This will show us the rules of the game.
Also we can see the control panel here.
The keys we will use to play the game are shown here.

2. Pause game:

When playing the game if we want to pause the game and play the game later we need to click the key “Esc” and the game will be paused and the main menu will reopen.From there we can access the game where we left it before and play the game by clicking the same key “Esc".It's a very useful option because no one can play the game continuously ,so if we want to leave the game in the middle we can use this option.

3. Game Over Features:

This page will show up after you win the game or run out of lives. The name of the player will be requested. The participant has to enter their name in 10 characters. The player must press "Enter" after entering their player name. Following that, a new page with the player's score and payer's name will emerge. The player is then directed to the main menu screen.

• Game-Play Features:

1. Multiple Levels:

DX-BALL has a total of three different levels. The difficulty of those increases proportionally. Details about the game’s levels are given below:

1.1 Level 1:

This level contains only single hit bricks. Various types of power up will appear in this level randomly. This level's difficulty is much easier than other levels. As this is the first level of the game we tried to make it easier for the users.

1.2 Level 2:

Both single-hit and multiple-hit bricks are present in this level. In this level, several power ups will also occur at random.The challenge of this level is greater than level 1. We made an effort to make the game's second level more challenging for the players.

2. Special Bricks:

Some special bricks are used in the game to increase the difficulty of level-2 and level-3 which will increase the difficulty of those level .Those are:

2.1 Multi Hit Bricks:

These bricks will appear in levels 2 and 3 and require two hits to eliminate.

2.2 Invisible Bricks:

The third level will have these bricks.These bricks will become apparent after the ball strikes them for the first time.

3. Sound Effects:

In this game, several sound effects are present. When the ball strikes the paddle, the ball or fire hits the bricks, a life is lost, a power-up is obtained, the laser paddle begins to fire, or a new level is begun, these sound effects will play. The "Option" menu's "Sound" icon can be disabled from the main menu page or the pause menu page if a player wants to disable the sound effects.

4. Score and Life Indicator:

Score and Life Indicator: After the game begins, the player can view the score and life indicator separately in the top left and top right corners of the page (Figure 8). The SDL TTF is used to create the score indication (True Type Font). The score indicator will display the final result. A life indicator, on the other hand, will indicate how many lives (paddles) are still available. These indications will continuously update.

Project Module:

The total game code is divided into multiple header files and source code files. In this game we have implemented 10 custom header files. Details about those custom header files are given below:

1. #include "pre_declare.h"

All pre-processors used in the code are listed in this header file. Additionally, it includes the flags for the renderer, window, and game over. It also includes all of the level backdrop textures and a few significant global variables with extern.

2. #include "sound.h"

This header file contains all the sound related variables and functions.It also contains some sound related boolean variables which are used globally in the code. Details of the functions:

2.1 void music_load();

The game's music and sound effects are loaded using this function. such is the main menu music, the expansion paddle music, the game over music, the bar paddle impact sound effect, and so forth. This function has all of the power-up-related music loaded.

3. #include "struct.h"

All of the structs that we utilized in the code are declared in this header file. In the entire code, we have utilized various struct types (s).These strut(s) are utilized for a variety of tasks, including rendering paddles, storing power-up information, and storing brick coordinates. The variables or arrays for these structs are likewise declared in that unique header file.

4. #include "utils.h"

The SDL libraries are initialised using this function. This user-defined function initialises the libraries for SDL, SDL ttf, SDL mixer, etc. This function only returns an integer and doesn't accept any parameters. It will return 1 at successful completion of the initialization procedure or 0 upon improper completion of the initialization process. The window and renderer are also created and initialised by this function.

4.2 SDL_Texture *DISPLAYING_SCORE(char ch[ ],int colour,int make_nu

This function is for printing the score. This function takes a string a variable named ”colour” and a flag as parameters and returns a texture.This function actually takes a string and make it a image using SDL TTF with specific colour and font which needed to be declare.

4.3 int levelup_bricks_initialization(int level);

This function is for every level's bricks distribution. In this function we have generated all the brick's coordinates using FILES. This function takes a variable which is the current level as a parameter and returns an integer which contains the total number of bricks in that specific level. In this function bricks distribution is made by using FILES.

4.5 void mainmenu_level_renderer();

This function is for rendering bricks for each level. At the starting of a new level this function is always called.

4.6 void ball_music_and_powerup_load();

This function loads the ball's image, some music of the game and all the power-up related images. The images we have used in the power-up indicator are also being loaded here.

5. #include "mainmenu_resources"

This custom header file contains all the UI related variables, texture and functions. More specifically, this header file has SDL Rect, Textures and functions which are related to the main menu page and pause menu page.
Details about the functions:

5.1 void mainmenu_load();

This function loads all the main menu related images and creates textures from surfaces. In this function we have generated the coordinates of those images. This function

also loads some pause menu related images and creates textures from surfaces too.

5.2 int mainmenu_render();

The main menu page is rendered using this function. This function has no input parameters and outputs an integer that can be either 0 or 1. The game returns 0 if it is closed from the main menu page, and 1 otherwise. This feature also directs the user to several menus, like the high score menu, the help menu, the option menu, the finish menu, etc. This function mostly displays the entire main menu page. Both the mouse and the keyboard can be used to navigate this page.

5.3 void pause_menu_renderer();

This function is for rendering the pause menu page. This function also takes a player to a different menu such as help menu, option menu , end menu etc. Mainly this function visualises the whole pause menu page. This page can be controlled with both mouse and keyboard.

5.4 void option_render();

This function renders the “Option” menu. This function controls turning on or turning off the music. Also game play sound effects turning on or turning off are controlled by this function depending on the user’s input.

5.5 void highscore_render();

This function renders the “High Score” menu which shows the top 20 high scores. This function takes the top 20 scores from a file and renders them in two pages.

5.6 void help_render();

This function renders the “Help” menu which shows another three pages named Instruction, Controls and Power-up. These sub pages are also being rendered from this function.

6. #include "gameplay.h"

This custom header file contains all the Game play related variables,textures and functions. These functions are being used for loading game play data and rendering them. Details about the functions:

6.1 void score_and_life_print(int score,int life);

This function is for printing the current score and lives that a player has. Actually this function is for creating and rendering the score indicator and life indicator in time of game play.

6.2 void bricks_and_bar_load();

This process loads the bricks and paddles. The game uses numerous different sorts of bricks and four different types of paddles. We are building their textures from the surfaces of

all those paddles and bricks as they are loaded from this location.

6.3 void firerender(int fbar);

This function is for rendering the laser. After achieving the laser paddle this function renders the laser from the corner sides of the bar depending on the player's input.

6.4 int bar_and_bricks_render(int cnt,int l);

This method renders the bricks and paddles based on their coordinates. In this function, the paddle's animation has been completed. The total number of bricks is returned by this function as an integer value with two integer parameters. This feature mostly depicts the paddles and bricks.

7. #include "game_physics.h"

This header file contains variables and functions related to the ball and paddle movement of the game. This part is the main part of this DX-BALL project. Details about the functions:

7.1 void game_physics();

This function controls the ball's and power-up's movement. This function contains the direction and vectors of ball's and power-up's. Also the ball's and grab paddle's combined movement is controlled by this function.We initially suppose that the ball will move at an angle of 30 degrees.

Here, the ball's coordinates' horizontal and vertical changes are denoted by x and y, respectively. Therefore, the modification to the ball's coordinate is,

θ0=30, v = BALL SPEED,

$$\Delta x = v \cdot \sin(\frac{\pi\theta_0}{180})$$

$$\Delta y = v \cdot \cos(\frac{\pi\theta_0}{180})$$

7.2 void collision();

The ball's and bricks' collision is managed by this function. Bricks will vanish after collision. Additionally, the laser and block collision is managed by this function. From here, power up starts are also made, and this is also when the collision sound effect begins.

7.3 void ball_fall_paddle_collision();

This function controls the fall effect of the ball and also controls
the paddle and ball collision. Bar and paddle collision sound effect and fall musics are also being controlled from here. The main part of the function is the ball's deflection in after hitting the paddle.
Here, we assume the angle to be θ for the ball's deflection. Here Δx is the horizontal change and Δy is the vertical change of the ball's coordinates,"ball.x" and "ball.y" is the x and y coordinate of ball, "paddle.x" and "paddle.y" is the x and y coordinate of paddle. So the change of the ball's coordinate is,

$$v = BALL\_SPEED, w = Paddle's\_width$$
$$\theta = \frac{(w - [ball.x - paddle.x])}{w} \cdot 180;$$
$$\Delta x = v \cdot \cos(\frac{\pi\theta}{180})$$
$$\Delta y = v \cdot \sin(\frac{\pi\theta}{180})$$

8. #include "powerup.h"

This custom header file contains power up related variables and functions. Details about the functions:

8.1 void powerup_achieve();

This function is for selecting which power up a player has achieved. This power up is generated randomly

8.2 void powerup_renderer(int p, int type);

This function is used to display the particular power up. The movement, velocity, and vectors of the power up are also included in this function. This feature controls the movement of the power up icon and mostly displays the attained power up icon.

9. #include "end.h"

This custom header file contains some functions which are used for resetting the game or freeing up the space before the game. Details about the functions:

9.1 void reset_game(int flag);

This function takes an integer variable as a flag. Basically this function is used for resetting the game when a player loses a life, is promoted to a new level or starts a new game.

9.2 void quit();

9.3 void font_closing();

9.4 void level_destroy();

These functions are used for freeing up the space by destroying the window , destroying the textures, closing the fonts and this function also closes the SDL, SDL TTF, SDL IMAGE , SDL MIXER etc libraries.

10. #include "header.h"

There are no variables or functions in this custom header file. Instead, it includes all of the custom header files as well as the library header files.We have only included this particular header in all custom header files and source code files because it contains all header files.

Team Member Responsibilities:

Team Member 1:

Tahshina Islam Munni,Roll-26

1.1 Paddle Animation:

This module includes the general paddle, grab paddle, laser paddle, and combination grab and laser paddle. Each paddle has a separate horizontal axis that it rotates around. The left mouse button can be used to release an electric force field that will catch balls that land on the grab paddle.

1.2 Paddle Movement:

Paddle can be controlled by moving the mouse and keyboard. If the player moves the mouse left/right, the paddle moves left/right.

1.3 Ball Movement:

The ball moves in accordance with the angle from which it is coming when it bounces off a paddle or collides with the top, right, or left corner of the screen. The side of the paddle the ball lands on determines the angle at which it will bounce off. The ball will be oriented more precisely in that direction the farther left or right the paddle is struck. Changes in ball speed are influenced by power ups.

1.4 Game level 1 and 2's Bricks Distribution:

The game contains three levels and different assemblies of bricks. The coordinates of the bricks are generated and then the bricks images are rendered in the selected coordinates.

1.5 Ball and Bricks Collision:

When the ball collides with brick,the brick will vanish and the ball starts to move to the opposite direction with the geometric angle which depends on its incoming angle. A sound will appear when the ball hits a brick.

Team Member 2:

Diponker Roy,Roll-28

2.1 Main Menu Page Design:

All of the functions connected to the main menu page, including producing the power-up pictures at the bottom of the screen and displaying new game, high score, settings, help, and exit buttons. A button's selection will direct the player to the associated page.

2.2 Score Print:

When the ball or laser collides with the brick or power up is achieved,the player gets 5 points.The score is shown in the top left corner of the screen.

2.3 Life Print:

The remaining paddles (lives) in the game are displayed in the top right corner of the screen.

2.4 Paddle Movement with Keyboard:

Paddle can be moved by right and left keys of the keyboard.

2.5 Game Level 2's Bricks Distribution:

In game level 3, the text "THE END" is created by various colours of bricks. First the coordinates of the bricks are generated and then the bricks images are rendered in the selected coordinates.

Team Member 3:

Mohammad Asif,Roll-54

3.1 All the Sound Related Works:

There are two sound types in this game. They are gaming sound and UI music. Those sound effects will come on at various points. By selecting the icon found in the "Option" section of the main menu page or the pause menu page, those can be played or disabled.

3.2 High Score Orientation:

In this section the scores are saved in a file sorted in descending order. When a new score is achieved , it will be compared with the previous data.If the score is greater than the lowest score, then all the scores will be sorted again and those scores will be saved in a file for future uses.

3.3 Game's Help and Instructions and Exit Section:

Controls, instructions, and a power-up button are all featured in the game's help section. The player will be able to learn more about the game's details by hitting those buttons. When a player clicks the exit button, a confirmation message will show up. If the player clicks OK, the message will close or the player will be taken back to the main menu.

3.4 Game over Page Presentation:

A game over screen with the text "Write your name" will appear when a player uses all of their lives or completes every level. The score will be displayed with the players' names after they have written them. The primary menu will then appear on the screen.

Platform, Library and Tools:

•Platform: Linux.

•Language: C/C++.

•Library: SDL (Simple DirectMedia Layer) library.

•Tools : Adobe Photoshop, Pics Art Photo Editor, Mp3 Cutter, Online Music Converter, Online Image Resizer.

•LaTex editor: Overleaf

•Video editor: OBS Studio, Kdenlive

Limitations:

While playing the game, a player could run into several restrictions.
•The DX-Ball project is designed for Linux systems.
•The player's name will only be allowed to contain 10 characters when it is stored for high score records.
Following a new start, all previous game records—aside from top scores—will be lost.

Conclusions:

We gained a lot of knowledge from working on this project, but we also encountered several challenges. For us, creating

a game is a completely novel and useful experience. As a result, we had to start from scratch. We have introduced the SDL (Simple DirectMedia Layer) library through this work. We are now aware of the creation and animation processes for models.Our capacity for thought and creativity has increased. We have improved our communication abilities through group collaboration. Actually, for us it was a completely new experience.

The challenge we've encountered is that creating a game is entirely foreign to us and differs from the programming we're used to. We had to start from scratch learning things that were entirely new to us. The internet, learning materials, and video tutorials have all helped us learn new things. Overall, it's a difficult task. It requires perseverance, time, and hard effort. Because we attempt to link the game environment with the actual world through the game, it can be claimed that creating games is a very sensible activity to engage in. It was challenging to deal with each and every aspect of the model to make it user-friendly.

At first, we believed it would be simple to create a DX-BALL game. But we later discovered that it's not that simple. since we had to do everything by ourselves. Everything has to be put into action by us. We had to do everything on our own, from moving the ball to keeping track of the score. We therefore had a lot of work to do, from the Ui to the game play. Therefore, the realization or experience went much beyond what we had anticipated. Overall, it was a wonderful and novel experience, and we are pleased to have incorporated our own suggestions and those of the teacher into our own codes.

Future plan:

•Improve the graphical representation of the game.
•Game level extension.
•Introduce and implement new game features.
•Take user feedback and improve features according to their opinion.
•Introducing a new environment and scenes.
•Make the game suitable for Windows and Mac OS platform.

References:

SDL Wiki: https://wiki.libsdl.org/Tutorials

Lazy Foo: https://lazyfoo.net/SDL tutorials/

Steam Community:
https://steamcommunity.com/sharedfiles/dx-ball

Lib SDL(TTF section):
https://www.libsdl.org/projects/SDL ttf/

Lib SDL(Mixer section):
https://www.libsdl.org/projects/SDL mixer/

GeeksforGeeks:
https://www.geeksforgeeks.org/write-header-file-c/

Mixkit:
https://mixkit.co/free-sound-effects/game/
Typesetting(Overleaf):
https://www.overleaf.com/read/jftsqwsdkmzj